Hands-on Learning of Hardware and Systems Security

Article *in* Advances in Engineering Education · April 2021

CITATIONS
3

READS
1,544

3 authors, including:

Shubhra Deb Paul
University of Florida
9 PUBLICATIONS **22** CITATIONS

SEE PROFILE

Hands-on Learning of Hardware and Systems Security

Shuo Yang, Shubhra Deb Paul, and Swarup Bhunia

Department of Electrical and Computer Engineering

University of Florida, Gainesville, FL, USA

Email: shubhra.paul@ufl.edu

Abstract

Hardware security is one of the most researched areas in the field of security. It focuses on discovering and understanding attacks and countermeasures for electronic hardware that provides the "root-of-trust" for modern computing systems upon which the software stack is built. The increasing reliance on electronic devices in our everyday life has also escalated the risks of experiencing security threats on these technologies. Students today are exposed to these devices and thus require a hands-on learning experience to be aware of the threats, solutions, and future research challenges in hardware security. Currently, there are limited opportunities for students to learn and understand hardware security. A significant factor limiting exposure to these topics is the lack of an accessible, low-cost, flexible, and ready-made platform for training students on the innards of a computing system and the spectrum of security issues/solutions at the hardware-level. In this paper, we introduce the motivation and efforts behind a course named "Hands-on Hardware Security." The Department of Electrical and Computer Engineering at the University of Florida has been offering this course for the past three years in providing experiential learning of hardware security through a set of well-designed experiments performed on a custom hardware module. We also present, in detail, the idea of a custom-designed, easy-to-understand, flexible hardware module with fundamental building blocks that can emulate a computer system and

create a network of connected devices. We refer to the module as “HaHa SEP” (Hardware Hacking Security Education Platform), and it encourages students to learn and exercise “ethical hacking,” a critical concept in the hardware security field. It is the first and only known lab course offered online, where students can perform ethical hacking of a computing system using a dedicated hardware module. This paper also provides a brief introduction to the experiments performed using this module, highlighting their significance in the field of Hardware Security. Finally, it concludes with a compilation of course evaluation survey results discussing the success of this course in engaging students’ interest in the subject matter and determining the accomplishment of maintaining a balance between their expectation and the effort required towards the course.

Keywords: Hardware and Systems Security, Hands-on Learning, Experiential learning

Introduction

What is Hardware Security?

Hardware and systems security has become an essential part of advanced computer science education in recent times. As a result, such a growth of cyber-infrastructure in modern society has escalated the need for its security. Hardware security elucidates the study of vulnerabilities and countermeasures in the architecture, implementation, and validation of modern electronic systems. It has evolved in parallel along with hardware design, and it forms an integral component in computer security research (Bhunia *et. al.*, 2018; Tehranipoor *et. al.*, 2011). Compared to the study of software security, which has been analyzed and deployed in various applications, hardware security is relatively new because the hardware has traditionally been considered immune to attacks and hence formed the trust anchor or root-of-trust of a system. However,

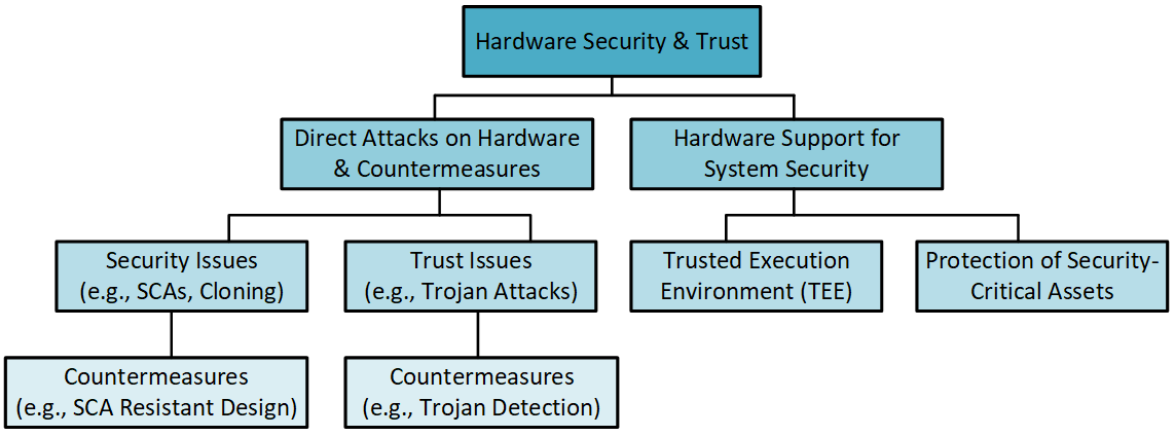


Figure 1. Scope of hardware security and trust for modern computing systems.

cyber-security experts and researchers have reported various security vulnerabilities and attacks on the hardware and embedded systems during the last thirty years

(Papp *et. al.*, 2015; Fournaris *et. al.*, 2017; Acohido, 2018; Robertson *et. al.*, 2018). Additionally, some of the recent researches have exposed a significant number of existing security vulnerabilities in computer systems deployed globally, such as the Rowhammer bug (Kim *et. al.*, 2014), RAMBleed (Kwong *et. al.*, 2020), Spectre, and Meltdown (Kocher *et. al.*, 2019; Lipp *et. al.*, 2018), ZombieLoad (Schwarz *et. al*, 2019), and many more.

Hardware security contains a wide range of topics, as depicted in Figure 1. The overall concept of hardware security and trust can broadly be categorized into two classes: direct attacks on hardware, including respective countermeasures and system-level security. The hardware attacks encompass both security issues and trust issues. Hardware security issues arise from its vulnerability to attacks (*e.g.*, side-channel or hardware Trojan attacks) at different levels of abstraction (chip or PCB) and the lack of adequate hardware support for software and system security. On the other hand, hardware trust issues arise when the untrusted entities get associated

with the hardware's lifecycle. These entities include not only untrusted IP or computer-aided design (CAD) tool vendors but also comprise untrusted design, fabrication, test, or distribution facilities. Another critical feature of hardware security is ensuring the security and reliability of the software stack. It protects sensitive assets stored in hardware from mischievous software and network and isolates secure data from insecure data and code. Additionally, it separates the applications among multiple users (Ray *et. al.*, 2018). There are two major topics in this area. The first one is the trusted execution environment (TEE) that protects the code and data of an application from untrusted applications. The final one is protecting security-critical assets in an SoC through an appropriate realization of security policies, such as access control and information flow policies, which govern the CIA (Central Intelligence Agency) requirements for these assets.

In the past, people focused on vulnerabilities in cryptographic chips leading to information leakage. However, in recent times, severe emerging security concerns have been recognized and studied. These include hardware Trojan attacks (Tehranipoor *et. al.*, 2010) in an untrusted design house or foundry, side-channel attacks (Zhou *et. al.*, 2005) where secret information of a chip can be extracted through measurement of side-channels (such as power, delay, and electromagnetic emission), IP (Intellectual Property) piracy and reverse-engineering attacks (Torrance *et. al.*, 2009) on ICs (Integrated Circuits), Modchip attacks and bus snooping attacks in printed circuit boards (PCBs). These attacks can occur at any point throughout the lifecycle of hardware components, from design to end-of-life, and encompass all abstraction levels from chips to PCBs to the system level. The complexity of these attacks illustrates the scope and need for hardware security as an emerging branch of research in hardware and systems security. The enormous scope also implies that college students would require experimental devices and tools capable of supporting a broad

range of applications for practice and flexible enough for potential future upgrades at a lower cost. Such devices would make it easier for students to learn and practice applying hardware security concepts.

Motivation behind Designing a Course on Hardware Security

There is a growing need for a real-world, flexible platform that would enable students, security researchers, and teachers to practice, train, and explore concepts that they learn and develop in the domain of "Hardware Security and Trust." At the time of this writing, over ten universities worldwide have been offering hardware security courses to their graduate and undergraduate students. A number of these universities offer these courses online, making it possible for students to learn hardware security remotely. However, our research indicates that no platform in the market allows students to apply the theoretical concepts beyond the classroom. Additionally, only a small amount of curricula include hands-on projects for the students. These projects also play a minor role in the composition of grades compared to their homework and examinations. Therefore, the need for hands-on experiences while learning the concepts for the students is not well addressed. To the best of our knowledge, ours is the first institution to offer a custom-designed platform that students can use for learning all aspects of hardware security.

Introduction to the Course

To address the emerging need for practical hardware security education with well-designed hands-on experiments, we developed the course "Hands-on Hardware Security." It provides the students with a hands-on learning experience on various existing attacks and threats on hardware at the chip and PCB level, which allows them to explore different countermeasures for some of these attacks. This practical learning element can play a pivotal role in realizing numerous

vulnerabilities related to a complicated hardware system's security aspects and necessary protective measures.

This course's backbone material is the custom-designed easy-to-use printed circuit board, titled the "HaHa SEP." The board can effortlessly mimic the functionality and complexity of a connected computer system. This platform's flexibility also enables students to apply all the aspects they learn about hardware security and practice and successfully fulfill the course requirements.

We craft the course experiments in a thought-provoking manner to pique the interest of the students. These experiments aim to motivate students to delve and explore Hardware Security concepts and develop innovative solutions to detect or prevent these vulnerabilities. In addition to the hardware design, the course offers students exposure to software development with well-refined exercises and assignments in popular programming languages such as C, C++, Python, Matlab, and Verilog to maintain balanced content.

The rest of the paper is organized as the following. The *Methodology* section describes the approach we used to build this course, including the steps involved in manufacturing the "HaHa SEP." We also describe the strategy behind the design of lab experiments and how we evaluate the efficiency of teaching the course. The *Results* section presents valuable student feedback, including their performances in the course and their evaluations. Finally, we conclude the paper with the corresponding section.

Methodology

This section discusses the course's detailed description, features of the HaHa SEP, and learning assessment techniques. We highlight the various aspects of the curricula that enable the students to achieve the right balance between the broad-theoretical scope and the challenging practical elements in hardware security.

Hands-on Hardware Security Course Description

This course focuses on the practical learning of different aspects of hardware security using a hands-on approach. The students get the unique opportunity to work on a custom-designed hardware platform, HaHa SEP, to understand a computer system's internal components and ethically "hack" into it at different levels. They examine it to understand security vulnerabilities, mount attacks and implement corresponding countermeasures.

Course Objectives

This lab course comprises a set of well-designed hands-on experiments that intends to help students realize the following.

- Understand the primary computer system security concepts, which integrate network and information security, software security, and hardware security.
- Learn about hardware components of computer systems and understand their security vulnerabilities through hands-on experience.
- Learn and design existing solutions and countermeasures against known attacks.
- Learn to hack into hardware ethically, devise new attack models and defense mechanisms against them.
- Analyze and validate computer hardware security issues and build a secure computer system.

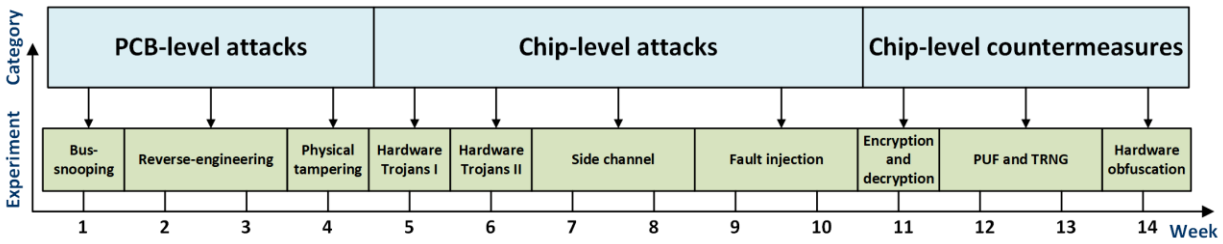


Figure 2. Timeline and progression of lab experiments.

Key Concepts Learnt in the Course

- Learn buffer overflow attacks – stack overflow, heap overflow, and array indexing errors.
- Learn about various hardware attacks at different levels – from chips to printed circuit boards (PCBs).
- Learn bus snooping attacks and protection schemes through bus encryption.
- Learn hardware tampering attacks (*e.g.*, Modchip attacks) in the field after deployment.
- Understand side-channel attacks, including fault injection and power analysis attacks, and hardware Trojan attacks of different forms and sizes triggered by rare events.

Understand various countermeasures against hardware attacks, including hardware authentication.

Reverse-engineer a printed circuit board (PCB) to explore and understand the interconnections and operations of onboard hardware components.

Course Structure and Schedule

We group the overall course materials into three major categories: (1) PCB-level attacks, (2) chip-level attacks, and (3) chip-level countermeasures. These three categories contain ten experiments in total, and it takes 14 weeks to complete all of them.

The complete timeline of the whole course is illustrated in Figure 2. Here, 6 out of 10 experiments need one week to complete/each, and the rest take two weeks/each. Usually, those four

experiments are substantially task-oriented and more time-consuming than others. The class meets once weekly, and each meeting lasts for 3 hours. Each session typically consists of a lecture on the topic and hands-on training on how to experiment.

In the classroom, the students are provided with the custom-designed hardware module (HaHa kit) at the beginning of the semester, along with a detailed instruction manual. Students borrow the kit throughout the semester and return it at the end. The provided manual contains in-depth instructions and examples on how to use this board for different purposes. For each experiment, students receive written instructions (uploaded on the course website) on the experiment objectives, steps to mount an attack or implement a countermeasure, parameters or waveforms to observe or demonstrate, and the reporting format. At the beginning of each new experiment, the instructor introduces the topic, steps of the experiments, advanced options (if any), and expected learning outcomes. For EDGE (Electronic Delivery of Gator Engineering) students, we post a video recording of this introductory lecture (15 min) in the Canvas, the campus-wide E-learning portal of the University of Florida. The assigned teaching assistants are available to help in-campus students in the lab and EDGE students via Skype/Zoom, as necessary.

HaHa SEP

HaHa SEP is the short form for “Hardware Hacking Security Education Platform.” It is a custom hardware module, which the authors designed to facilitate performing all the hands-on experiments for this course on a single platform. As the material of the course covers various

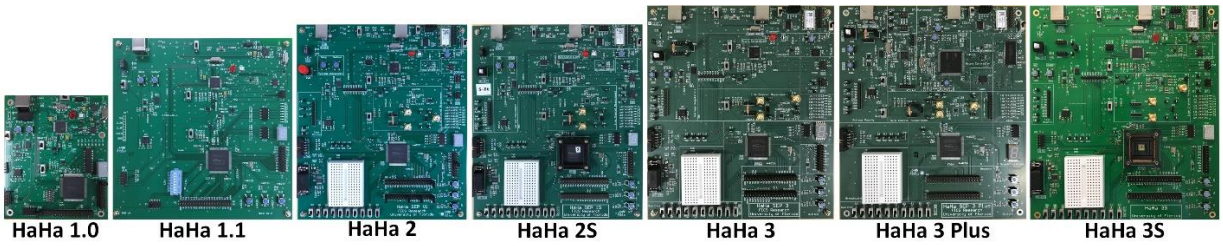


Figure 3. The design of the HaHa SEP board has been revised over the years to improve its capability to run diverse hardware security experiments. The picture shows three generations of the HaHa SEP board and their relative sizes.

aspects of hardware security, the experimental platform is equipped with numerous functionalities and features to maximize the number of supporting experiments. Figure 3 shows the evolution of the HaHa SEP board over several years. It has been updated over three generations to improve students' learning experience and to augment its flexibility to conduct numerous hardware security experiments on a single platform. These features are not collectively present in any commercially available Field Programmable Gate Array (FPGA)/microprocessor-based development boards or even in special-purpose platforms, *e.g.*, Sakura side-channel attack evaluation board. The existing Altera/Xilinx FPGA-based development boards are suitable for implementing only a few experiments, such as hardware Trojan attacks or primitive security design in boards or power side-channel attacks in Sakura board (Technologies; Xilinx; Sakura Project). Still, they are not amenable to implementing many others.

A photograph of the latest version of the HaHa SEP 3 (HaHa 3) board is shown in Figure 4, illustrating the board's configuration and highlighting all connectors and critical components' locations. The HaHa SEP core consists of two chips: an Intel MAX 10 FPGA and an Atmel 8-bit AVR microcontroller. These two chips are connected serially in the same JTAG (Joint Test Action Group) chain. The peripheral circuits of the FPGA include LED (Light Emitting Diode)

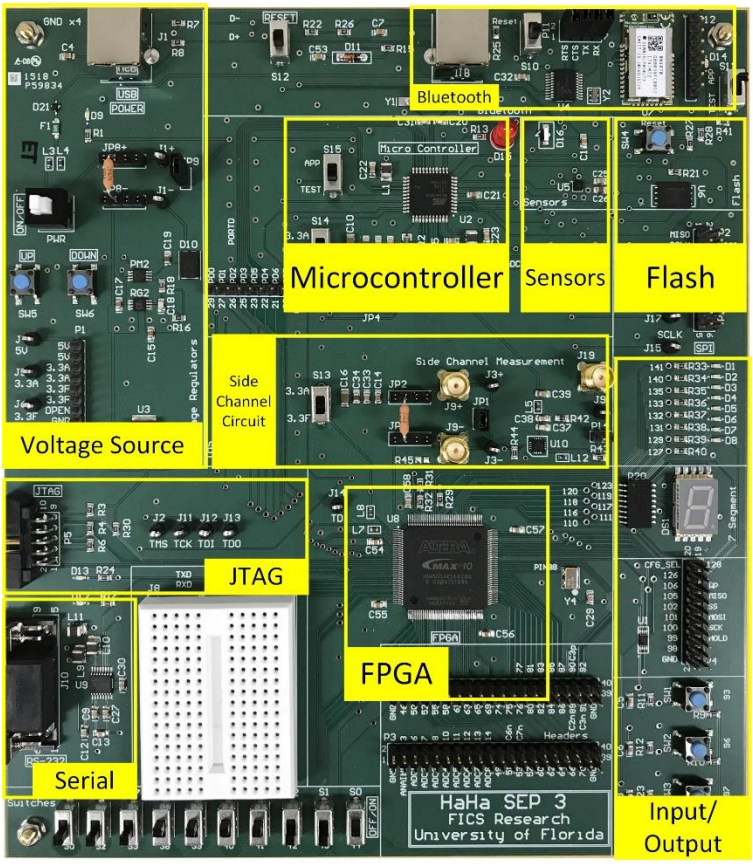


Figure 4. The HaHa SEP board and its major components. The board consists of two processing units: an Atmel microcontroller, and an Intel FPGA, sensors: light, temperature, accelerometer, switches, LEDs, I/O pins, side-channel circuits, Flash memory, communication channels: serial, JTAG, Bluetooth, and more.

indicators, a 7-segment display, pushbutton switches, breadboard, and headers, sensors, all of which can be used as general input and output approaches.

On the other hand, the microcontroller connects several SPI (Serial Peripheral Interface) devices, *e.g.*, an accelerometer, a Flash memory chip, and a Bluetooth module. Also, the FPGA and the microcontroller are interconnected through I/O (Input/Output) ports, making it possible for either of the chips to access all the components of the peripheral circuits on the board. The structure of the HaHa SEP is depicted in Figure 5. The key features which make the board suitable for the

course and various hardware security experiments are indicated in the diagram with red dashed boxes. We discuss all these available features in the upcoming sections.

For HaHa SEP 3, two alternative versions are designed and manufactured, and they are manifested in Figure 3: HaHa SEP 3 Plus (HaHa 3+) and HaHa SEP 3S (HaHa 3S). HaHa 3+ is an advanced version of the HaHa 3 that replaces the Atmel AVR microcontroller with a high-performance ARM Cortex-M4 32-bit microcontroller. The Cortex-M4 core features a floating-point unit (FPU) single precision, which supports all ARM single-precision data-processing instructions and data types. It allows students to do more software security experiments. On the other hand, the HaHa 3S board replaces the FPGA chip of HaHa 3 with an FPGA chip socket. The socket houses a 144-pin Intel Altera MAX 10 FPGA, and the chip can promptly be mounted or dismounted from the socket. This socket is surface-mounted on the board, with all the other components and configurations remaining unchanged. With this version of the board, students can perform specific experiments that require multiple FPGA chips, such as Trojan detection or the security primitive (PUF/TRNG) design experiments. These experiments often require multiple physical parameters measurements over several chips to evaluate the impact of process variations, environmental parameter variations, and aging.

Salient Features of the HaHa SEP

The HaHa SEP board has well-defined configurable user expansion ports, making it very flexible for students to expand its functionality. As exhibited in Figure 5, it equips three user expansion headers connected to the FPGA, two SPI (Serial Peripheral Interface) slots coupled to its corresponding microcontroller interface, and a breadboard to mount other chips and implement a custom circuit by the user. These headers, peripheral slots, and breadboard use standard

components compatible with commercially available connectors, wires, and chips that are not included in the HaHa SEP board.

Students can also reconfigure the chips' connections via several accessible onboard pins. Also, other peripherals, such as the Bluetooth module, have multiple reconfigurable pins. The board users have the freedom to change how the components communicate and work with each other. Two or more HaHa boards can be connected using a wired or a wireless connection mechanism in various configurations to build an interconnected system. The headers, probing points, and breadboards allow a connection of two boards using jumpers and wires. On the other hand, the Bluetooth module enables each HaHa board to communicate and control another. The ability to reconfigure the HaHa boards into a connected system offers many experimental options for the students.

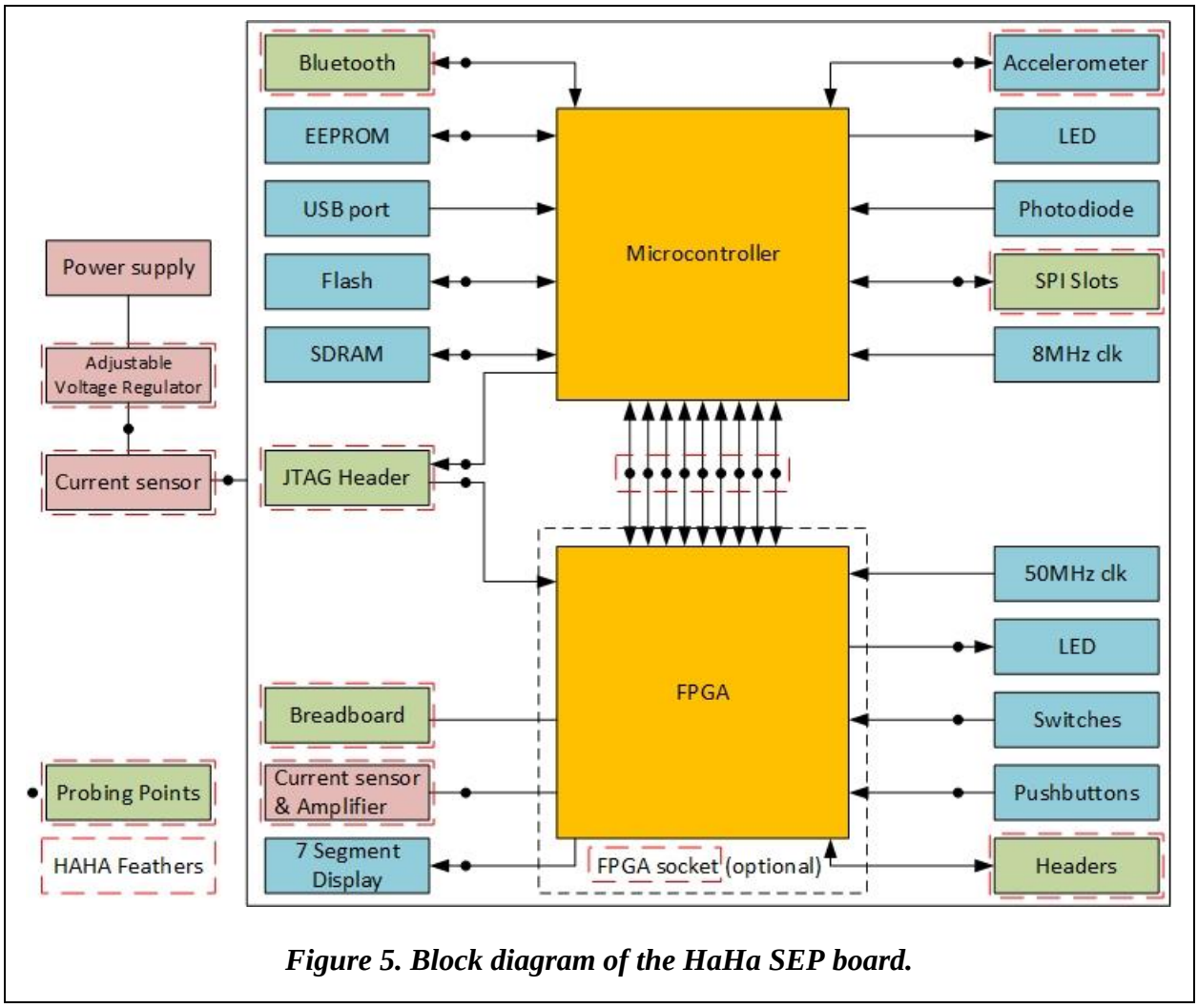


Figure 5. Block diagram of the HaHa SEP board.

We summarize these features of the HaHa SEP board in Table 1. We observe that the board is intentionally designed in a way that the students can apply all the experiments provided by this course on it.

Table 1. A selected list of features of the HaHa board

Feature description	Benefit for security experiments
The board includes both FPGA and microcontroller	Helps to implement diverse types of a computing system using either component or both.
FPGA and microcontroller both have embedded flash and are easily programmable using JTAG.	Both devices are easy to program. FPGA can act as an accelerator for the microcontroller.
FPGA and microcontroller communicate with the non-volatile memory	It helps to build an independent computing system with its memory to store configuration and input data.
Current/power monitoring for FPGA and microcontroller	Helps to run side-channel analysis and attack experiments.
Current/power monitoring for the entire board (in HaHa 3.0 and HaHa 3.0 Plus)	Side-channel analysis and PCB authentication.
Voltage control of FPGA using a potentiometer	PUF experiments with robustness analysis and fault injection attacks with power glitch.
Voltage control of microcontroller using a potentiometer	PUF experiments with robustness analysis and Fault injection attacks in embedded software with power glitch.
JTAG chain for FPGA and microcontroller	JTAG programming, JTAG Testing, and JTAG attack experiments.
Two-layer board	Facilitates reverse engineering experiments.
Simple layout with clearly marked regions	A clear understanding of the board design and configurations, and reverse engineering attack.
Multiple probing points on buses	Bus snooping attack and PCB tampering experiments.
Embedded breadboard and user headers	User's flexibility to include additional components and a small custom circuit on the board.
Configurable Bluetooth module	Bluetooth connection between boards to build a connected system and Bluetooth attack experiment.
Embedded temperature sensor in the FPGA	Temperature-triggered Trojan attack experiment, PUF/TRNG reliability & side-channel experiments.
Plenty of configurable headers	Physical tampering (Modchip) attack experiment.
Clock source and PLL inside the FPGA	Fault injection attack using the clock.
Integration of expansion headers	Flexibility to add more components or sister boards to HaHa to expand its functionality.

The HaHa Kit

We offer this course online, and it has also been made available to the EDGE (Electronic Delivery of Gator Engineering) students for the past three years. We provide the students with the HaHa Kit, and thus they are allowed to use the HaHa SEP board to perform all experiments off-campus. The provided kit comes with an Analog Discovery portable oscilloscope from Digilent, which can be used with a computer through a USB port to collect digital and analog signals from the HaHa SEP (Digilent).

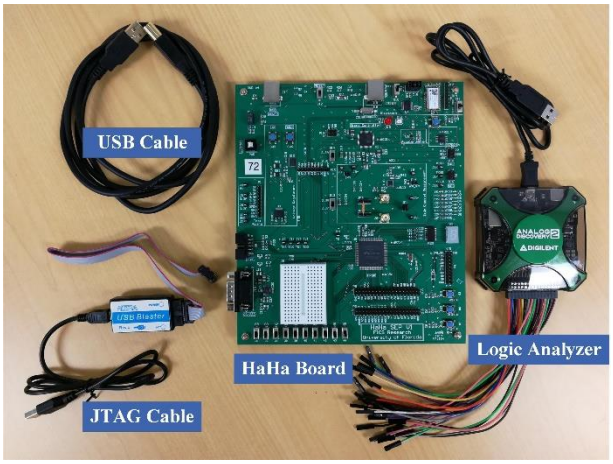


Figure 6. The HaHa kit provides a unique self-contained platform for hardware security training and education. The kit is suitable for the online offering of a hardware security lab course, where the students can acquire the kit from various possible sources and perform all the experiments at home without the need for a physical lab or special benchtop equipment.

Therefore, students need not be physically in a lab with electronic devices and instruments to practice all the experiments and finish the course in a take-home fashion. Figure 6 manifests the HaHa kit components, including the HaHa SEP board and a portable oscilloscope/logic analyzer, and necessary cables.

Lab Activities

This section briefly discusses the scope of the experiments performed in this course.

Experiment 1: Bus Snooping Attacks

Bus snooping is a technique used in distributed shared memory systems and multiprocessors to extract secret information (such as encryption key, firmware, sensed data) through physical access. This experiment aims to carry out a bus snooping attack at the board level, which leads to the retrieval of secret information from a system through physical access. A bus snooping attack can be implemented with no help from the software portion of the system. It involves eavesdropping or even altering data that are transferred between the two or more components of a system. Usually, there are numerous traces in a PCB, and other wires and ports transmitting signals,

which may contain confidential data or commands, are crucial to the system operation. Snooping into the traces by soldering additional components or wires allows the attacker to leak those sensitive data. The system can be forced to shut down or do something malicious just by altering the critical signals. The original Microsoft Xbox was initially hacked in the same way. The link between the Southbridge and EEPROM on the Xbox was observed during boot up and was modified to allow the hackers access to the protected region of the Xbox's hard drive (Huang, 2003). In mobile devices, the successful snooping attack may allow the attacker to interfere with data between SoC (System-on-Chip) and DRAM (Dynamic Random-Access Memory) or SoC and NAND Flash. Furthermore, the adversary can capture as well as alter code and data written from SoC to memory.

In this experiment, the students require to program the onboard microcontroller and observe waveforms from the data bus using an oscilloscope. By analyzing the waveforms, students can decode the communication protocol and determine what signals are getting transmitted.

Experiment 2: Reverse-Engineering

We design this experiment to understand reverse-engineering attacks, which are used to realize a hardware IP (Intellectual Property) and potentially clone it, leading to piracy or counterfeiting a product. Reverse-engineering, also known as back-engineering, is the process of extracting knowledge or design information from any human-made products and reproducing it exactly or reproducing relatively anything based on the extracted information (Quadir *et. al.*, 2016). The process often involves disassembling something (a mechanical device, electronic component, a computer program, or biological, chemical, or organic matter) and analyzing its components and workings.

The instruments needed for this experiment are the HaHa Board and a multimeter. In order to copy an IP, one first needs to understand how it works. One way of achieving this is by generating a schematic and a bill of materials (BoM). These can be accomplished by visual inspection or probing points on the board with a multimeter set to the 'Continuity' mode. It will cause a loud beep whenever the probes make an electrical contact together. After exploring the mutual interconnections between components, it is possible to generate the BoM by looking up the components' datasheets.

Experiment 3: Physical Tampering of Hardware (Modchip Attack)

The goal of this experiment is to perform a physical tampering/Modchip attack on a key-protected system. Modchip is a small electronic device used to replace, disable, or override the software protection of computers or entertainment devices such as video game consoles, DVD players, and TV cable-boxes. An adversary can introduce various modifications to its host system's functions, including the circumvention of region coding, digital rights management, and copy protection checks to use the media intended for other markets, illegally copying media, or using unlicensed third-party software.

In this experiment, we instruct the students to design a simple system, such as lighting up an LED. At first, an individual student plays the manufacturer's role, where he/she will store a limited key into the system ROM so that the function is inaccessible. Next, they will act as hackers and apply Modchip attacks to bypass the system's key protection mechanism and make it functional.

Experiment 4: Hardware Trojans I

Hardware Trojans refer to certain malicious modifications in the design of an IC or PCB by untrusted parties. They are very similar to software Trojans that attack operating systems or application software. This experiment intends to provide hands-on experience to the students

about hardware Trojan attacks in the form of malicious modifications of electronic hardware that pose major security concerns in the electronics industry. The emerging trend of outsourcing the design and fabrication services to external facilities as well as increasing reliance on third-party IP cores and electronic design automation (EDA) tools makes integrated circuits (ICs) increasingly vulnerable to hardware Trojan attacks at different stages of their lifecycle (Bhunia *et. al.*, 2014). Every party associated with the design and fabrication of an IC can be a potential adversary who can tamper with it. Such tampering can be accomplished through the addition/deletion/alteration of circuit structure or through modification of manufacturing process steps that cause reliability issues in ICs. From an attacker's perspective, the objective of such attacks can vary, *e.g.*, to malign the image of a company to gain a competitive edge in the market; to disrupt major national infrastructure by causing a malfunction in electronics used in mission-critical systems, or to leak secret information from inside a chip to access a secure system illegally. In this experiment, the students implement combinational and sequential hardware Trojan attacks on a 64-bit DES (Data Encryption Standard) algorithm. DES is a prominent cryptographic encryption algorithm, invented initially to secure sensitive government information from adversarial attacks.

Experiment 5: Hardware Trojans II

This experiment requires the students to design their hardware Trojans and perform attacks by triggering them through temperature sensor measurement signals from the MAX 10 FPGA of the HaHa board. This FPGA features one analog-to-digital converter (ADC), which provides a built-in capability for on-die temperature monitoring and external analog signal conversion. The temperature sensing mode monitors external temperature data input with a sampling rate of up to 50 K (thousand) samples per second (KSPS).

In terms of the payload, the Trojan can cause functional failure upon triggering or have a passive effect such as heating of the die or leaking of information (Bhunia *et. al.*, 2014; Roy *et. al.*, 2015). A Trojan can cause an "information leakage" attack, where a Trojan leaks secret information via a transmitted radio signal or serial data port. It could also involve a side-channel attack where the data is leaked through the power trace or thermal radiation or optical modulation of an output LED. Another type of Trojan payload would be an unauthorized alteration in system behavior. The instruments needed for this experiment are the HaHa Board, a USB Blaster, a computer, and a heater/hairdryer.

Experiment 6: Side-Channel Attacks

In cryptography, a side-channel attack is any attack based on information gained from the physical implementation of a cryptosystem, rather than brute force or theoretical weaknesses in the algorithms (compare cryptanalysis). For example, timing information, power consumption, electromagnetic leaks, or even sound can provide an additional information source, which can be exploited to break the system.

In this experiment, the students implement two different kinds of side-channel attacks to retrieve the secret key: Simple Power Analysis (SPA) and Differential Power Analysis (DPA) on DES. In cryptography, a side-channel attack is any attack based on information gained from the system's physical implementation. SPA is a technique that involves directly interpreting power consumption measurements collected during cryptographic operations (Zhou *et. al.*, 2005). SPA can yield information about a device's operation as well as a critical material. In addition to a large-scale power variation due to the instruction sequence, there are effects correlated to data values being manipulated. These variations tend to be smaller and are sometimes overshadowed by measurement errors and other noise. In such cases, DPA is deployed to break the system using

statistical functions tailored to the target algorithm (Kocher *et. al.*, 2005). As part of these lab activities, students implement SPA and DPA on the DES algorithm and use Matlab software to process the collected power trace data.

Experiment 7: Fault Injection Attacks

Fault injection attacks intentionally cause errors in a system to compromise the security of the system. This experiment targets to compromise the security of a system by implementing a Fault Injection attack. Fault injection techniques are developed to alter the correct functioning of a computing device maliciously. These approaches are achieved by varying the power supply voltage level, injecting irregularities in the clock signal, introducing radiation or electromagnetic (EM) disturbances, overheating the device or exposing it to intense light, *etc*. (Barenghi *et. al.*, 2012). In this experiment, students implement a 128-bit Advanced Encryption Standard (AES) algorithm on the FPGA of HaHa board and perform Differential Fault Analysis (DFA) on AES (Giraud *et. al.*, 2004). DFA is a type of side-channel attack in the field of cryptography, specifically cryptanalysis (Biham *et. al.*, 1997). The principle of DFA is to induce faults (unexpected environmental conditions) into cryptographic implementations to reveal their internal states. Nowadays, this technique is frequently used to test the security of cryptographic smart card applications.

Experiment 8: Information Security: Encryption/Decryption

In cryptography, encryption is the process of encoding messages or information so that only authorized parties can read it. Decryption is the process of taking encoded or encrypted text or other data and converting it back into text that a user or the computer can read and understand. This experiment aims to learn how to encrypt data with standard encryption algorithms, with a focus on using cryptography for protecting sensitive data with a secret key. In cryptography,

encryption is the process of encoding messages or information so that only authorized parties can read it. Decryption is the process of taking encoded or encrypted text or other data and converting it back into plaintext that the user or the computer can read and understand. In this experiment, the students learn about the Caesar cipher as one of the simplest forms of encryption and the AES algorithm. The Caesar cipher is also known as a shift or substitution cipher where the original message (plaintext) is replaced (ciphered) with another letter corresponding to a certain number of letters up or down in the alphabet.

On the other hand, AES is based on a design principle known as a substitution-permutation network, a combination of substitution and permutation, and is fast in both software and hardware. In our coursework, students learn both software and hardware implementation of the AES algorithm in Exp. 7 and 8.

Experiment 9: Hardware-based Security Primitives and Their Applications

This experiment's objective is to introduce two of the essential hardware-based security primitives: Physical Unclonable Function (PUF) and True Random Number Generator (TRNG), along with their applications to defend against counterfeiting attacks at the chip level.

PUF is one of the emerging potential security primitives for generating volatile secret keys in cryptographic applications (Gassend *et. al.*, 2002). A PUF is described as unclonable because its uniqueness is derived from the manufacturing process’s uncontrollable variations. When an external stimuli/input (challenge) is applied to a PUF, it generates a corresponding output (response). As a result, a PUF operation depends on these challenge-response pairs (CRPs), which are also known as signatures. These signatures determine the quality of the security and protection of a PUF.

PUFs offer a high level of protection in cryptographic applications with robust volatile key storage. In this experiment, students implement two of the most popular PUFs: SRAM-based (Static Random Access Memory) PUF and Ring-Oscillator (RO)-based PUF on FPGA and evaluate the performance on four categories: reliability, uniqueness, robustness, and randomness (Holcomb *et. al.*, 2009; Suh and Devadas, 2007; Maiti *et. al., 2013*, Zhang *et. al., 2020*).

On the other hand, a random number generator (RNG) is a vital security block widely used in most cryptographic applications such as one-time pads, session, and temporary keys, hardware metering, generation of primes, secure communications, secured servers and processors, virtual private network (VPN) access, and customer-facing web access (Majzoobi *et. al.*, 2011). A quality RNG generates statistically independent and unpredictable sequences of random numbers. Compromising an RNG is often related to compromising an entire system. A true RNG (TRNG) translates random physical phenomena such as thermal noise, atmospheric noise, flicker noise, clock jitter, phase noise, *etc.*, into random digits. In the second part of this experiment, the students modify the previously designed SRAM PUF and RO-PUF design to perform them as TRNGs and evaluate their performance through the NIST (National Institute of Standards and Technology) Test Suite (Rukhin *et. al., 2001*).

Experiment 10: Hardware Obfuscation

Hardware obfuscation is a method to hide the logic in a circuit design. By definition, obfuscation is the technique of obscuring or hiding the true meaning of a message or the functionality of a product to protect its inherent intellectual property. This experiment intends to implement different Hardware Obfuscation techniques for securing hardware IPs against piracy and tampering attacks. Hardware obfuscation is the technique to hide or obscure the actual logic or the functionality of a product to protect its original IP (Roy *et. al.*, 2010). The main objective of

hardware obfuscation is to have a functionally equivalent but structurally different design (Forte *et. al.*, 2017). The design is modified to implement different logic functions, so it is not possible to retrieve the correct logic equation by reverse engineering. A locking mechanism must be incorporated, ensuring the design becomes functionally equivalent to the proper unlocking process. In this experiment, the students learn to perform both *combinational* and *sequential* obfuscation and apply *brute force* attacks to break the FPGA's obfuscation using the Verilog *netlist* and *testbench* in the Intel Quartus platform.

Learning Assessments

The students are assessed based on a report for each experiment, along with an in-lab demo or remotely recorded demo. Students are expected to document the experiments' details in the lab reports, including the goal, experimental setup, and necessary steps. Also, in the experiment descriptions, students are instructed to answer multiple questions as a part of their assessment. Besides, on-campus students need to perform in-lab demos to the teaching assistants (TAs) to demonstrate the experimental results. The off-campus EDGE students need to record a video demo and submit it online with their experiment report. In addition to this, each of the experiments includes a few follow-up exercises, and we award the students with bonus points if they answer them correctly. The written report makes up 80% of the total points, and the demo takes up the remaining 20%. Bonus points usually contribute to another 10% of the overall grades. The final grade of each student comes from all the grades from the ten experiments.

Course Evaluation Survey

To determine the success of the learning process, we conduct surveys at the end of every semester. This general university-wide survey aspires to improve the quality of the instructors' learning, development, and evaluation based on the students' feedback received anonymously. The study

comprises ten questions, encompassing the success of this course into engaging students' interest in it and determining the accomplishment of maintaining a balance between their expectations compared to the required effort. Students are asked to select an integer score from 1 to 5 to choose during the survey for the questions, *e.g.*, "the quality of the contents to stimulate your interest in the course," "encouragement of independent, creative, and critical thinking," *etc*. We define the scores as:

1 - Poor/Low; 2 - Minimal; 3 - Moderate/Satisfactory; 4 - Above average; and 5 - High/Excellent.

Results

Survey Results

As this course is offered every Fall semester, the collected results for Fall 2017 and Fall 2018 semesters together are summarized in ***Table* 3**, where a total of 31 students participated. ***Table* 2** summarizes the demographic information of the survey participants. The demographic distribution of the survey participants approximately follows the department-level and college-level trends of such. Besides the course-wise mean score, the survey also generates both department-wise (Electrical and Computer Engineering) and college-wide (Herbert Wertheim College of Engineering) mean scores for that specific semester. The comparison of these three types of scores is depicted in Figure 7. From the figure, it appears that in most cases, the mean score for this course was higher than the department-wide and college-wide mean scores. We can conclude that the course encourages the students' engagement and provides them with state-of-the-art knowledge to prepare them for the industry.

Table 2: Demographic information of the survey participants.		
Criteria		**Percentage of survey sample**
Gender	Male	72%
	Female	28%
Race/Ethnicity	Asian/Asian American	66%
	Black/African American	4%
	Hispanic/Latino	6%
	White	22%
	Others	2%

Table 3. Combined results of the course evaluation survey from a total of 31 participating students in the Fall 2017 and Fall 2018 semesters.

No.	**Questions**	**Score**					**Mean Score**
		1	**2**	**3**	**4**	**5**	
1.	The quality of the description of course objectives and assignments	0%	0%	10.34%	24.14%	65.52%	4.55
2.	The quality of the contents to stimulate your interest in the course	3.33%	0%	6.67%	16.67%	73.33%	4.57
3.	The quality of the contents to facilitate the learning process	3.33%	0%	6.67%	23.33%	66.67%	4.50
4.	Enthusiasm for the subject	3.33%	0%	0%	16.67%	80%	4.70
5.	Encouragement of independent, creative, and critical thinking	0%	3.33%	3.33%	13.33%	80%	4.70
6.	Amount learned	0%	0%	12.90%	22.58%	64.52%	4.52
7.	Amount of effort required	0%	0%	9.68%	29.03%	61.29%	4.52
8.	The difficulty of the subject matter	0%	0%	25.81%	32.26%	41.94%	4.16
9.	The educational value (relevance) of this course	3.23%	0%	0%	22.58%	74.19%	4.65
10.	Expected grade	0%	0%	3.23%	9.68%	87.10%	4.84

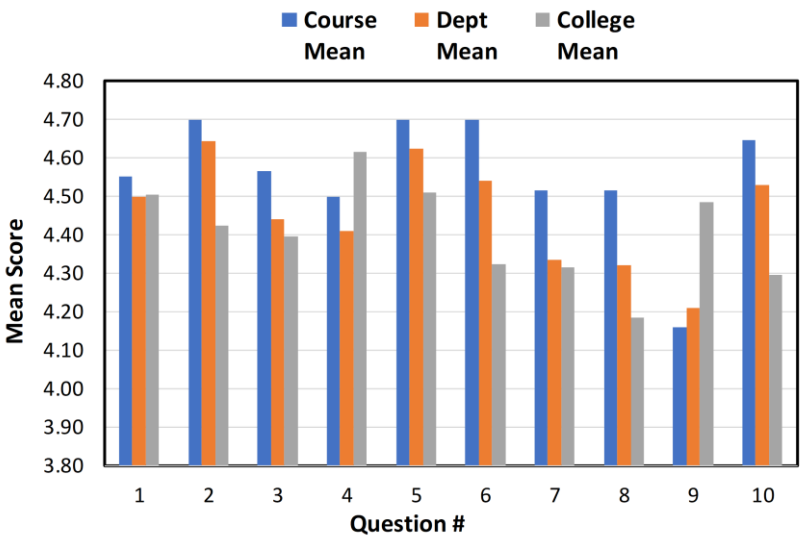


Figure 7. Comparison of evaluation results with department-wide and campus-wide mean value.

Figure 8 depicts the percentage score distribution for each of the ten questions. We observe that a significant portion of the students gave scores of 4 or 5 for those questions. It indicates that the course was successful in making a substantial impact on the students in different categories.

By close observation, we can divide the survey questions into two categories: Qs. 1-5 discusses the course's success in growing the overall interests of students plus the quality of the course contents and Qs. 6-10 are about their effort and expectations from it. The students' interest/attention and the course quality is roughly measured using the four following groups: (a) stimulating interest (Qs. 2), (b) overall learning experience (Qs. 1 and 3), (c) enthusiasm on the subject (Qs. 4), and (d) creativity (Qs. 5). Over 73% of students found that the contents could stimulate their interest in the course (Qs. 2). 66.67% of students thought it was a great learning experience for them (Qs. 3). Moreover, 80% of the participating students found it

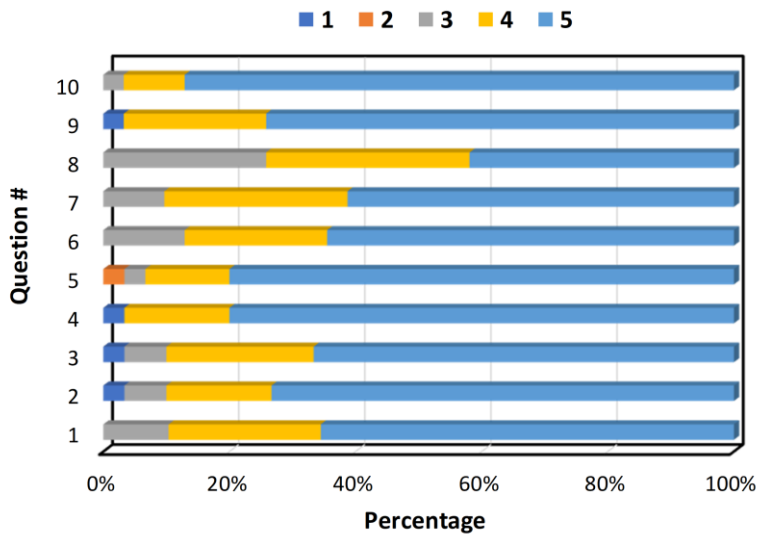


Figure 8. Percentage score distribution of 10 survey questions.

enthusiastic and felt that the learning process invoked their critical thinking and made them more creative (Qs. 4 and 5). Figure 9 elucidates these results.

Besides creating top-quality content, the success of a course also depends on the overall educational value and how it would impact the students in shaping their future, especially in

Engineering studies. The next set of 5 questions in *Table 3* (from 6 to 10) are prepared to evaluate: (1) amount learned, (2) effort required, (3) level of difficulty, (4) educational value, and (5) expected grade. More than 60% of students thought that the course requires a significant effort (Qs. 7). However, almost 65% of them mentioned that they learned a lot from it (Qs. 6). This course's difficulty level has received mixed ratings as more than 41% of students found the course very difficult, where 32% thought that it is in between moderately severe and 25% had a neutral view towards it (Qs. 8). As discovered from the survey, 87% of students were delighted with their final grades (Qs. 10). More importantly, almost 97% of students gave a score of 4 to 5 on this course's educational value (Qs. 9). It means that the course successfully provided high-

quality content to prepare the students for their future career goals related to the hardware security field. Figure 10 manifests the outcomes of these specific survey questions.

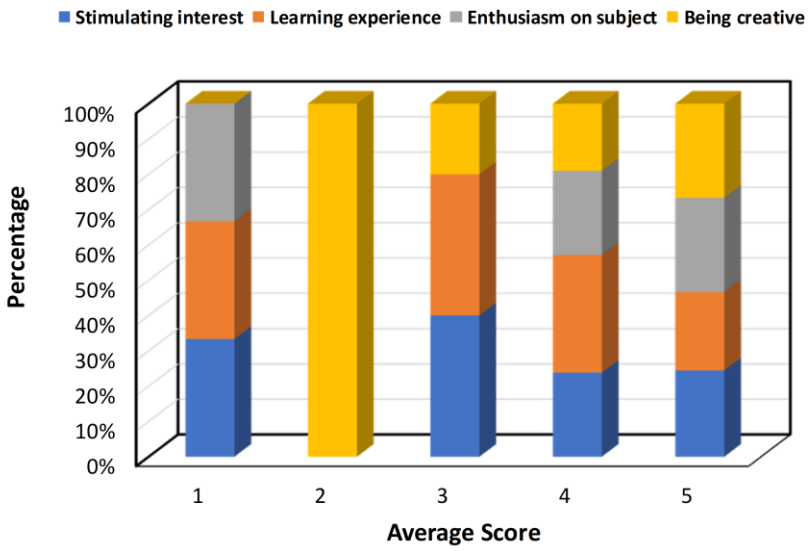


Figure 9. Illustration of students' interests, enthusiasm, and experiences.

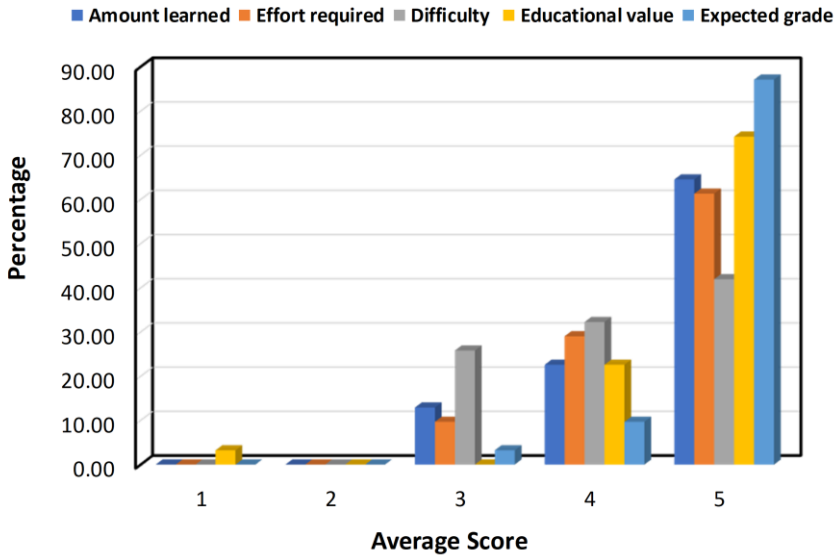


Figure 10. Educational value and outcome.

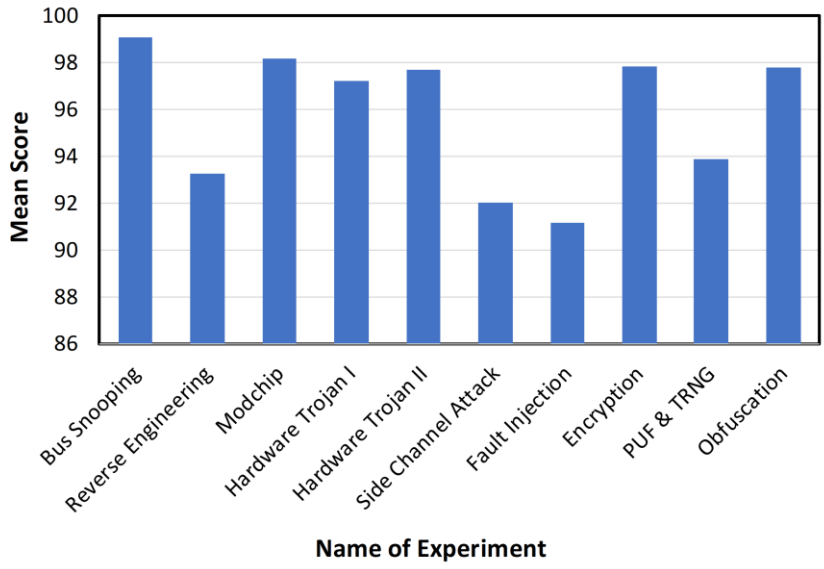


Figure 11. Average student scores for each of the experiments.

Learning Assessment Results

To identify the effectiveness of the lab experiments, we measure the students' performance by combining the score of the experimental report and demonstrations. The total grade point for each of the labs is 100. Figure 11 illustrates the average assessment results for each lab. In all the ten experiments, students achieved an average score above 90. Among them, the side-channel attack and the fault injection attack are the most difficult ones; and the bus-snooping attack, as the first experiment, is the easiest one. Overall, all the students performed exceptionally well in these experiments.

Conclusion

Due to the rapid growth of Hardware Security and Trust related research, the importance of practical experience beyond theoretical knowledge on real-world hardware attacks and corresponding solutions for the students and professionals is ever-increasing. To address this challenge, we have developed a 14-week course named "Hands-on Hardware Security." This

paper has discussed the various aspects of this course, including the design and development of a flexible and easy-to-learn hardware module called “HaHa SEP.” This well-balanced course incorporates ten different experiments to invoke students’ interest in various hardware vulnerability issues and perform research on possible countermeasures, and offer them the opportunity to nurture their software skills with computer programming exercises. Moreover, this paper reviewed the compilation of the results from course evaluation surveys from the Fall 2017 and Fall 2018 semesters, encompassing the course’s success in engaging students’ interest and determining its accomplishments in maintaining symmetry between their expectations and the effort required towards it. From the surveys, we found that the course was able to evoke the enthusiasm of more than 80% of participating students, and 87% of the students were delighted with their final grades. More importantly, approximately 97% of the students rated this course’s contents as very high quality in preparing them for their career aspirations in the field of hardware security.

Acknowledgment

This project is sponsored partially by the National Science Foundation (NSF) Award# 1623310, “EDU: Collaborative: HACE Lab: An Online Hardware Security Attack and Countermeasure Evaluation Lab,” PI(s): Mark Tehranipoor and Swarup Bhunia.

References

Acohido, Byron. LW’s NEWS WRAP: Meltdown, Spectre discovered in the wild – live hardware attacks one step closer. https://www.lastwatchdog.com/news-wrap-up-meltdown-spectre-discovered-in-the-wild-live-hardware-attacks-one-step-closer/. 2018

Barenghi, Alessandro, Luca Breveglieri, Israel Koren, and David Naccache. “Fault injection attacks on cryptographic devices: Theory, practice, and countermeasures.” *Proceedings of the IEEE* 100, no. 11 (2012): 3056-3076.

Bhunia, Swarup, and Mark Tehranipoor. Hardware Security: A Hands-on Learning Approach. Morgan Kaufmann, 2018.

Bhunia, Swarup, Michael S. Hsiao, Mainak Banga, and Seetharam Narasimhan. “Hardware Trojan attacks: threat analysis and countermeasures.” *Proceedings of the IEEE* 102, no. 8 (2014): 1229-1247.

Biham, Eli, and Adi Shamir. “Differential fault analysis of secret key cryptosystems.” In *Annual international cryptology conference*, pp. 513-525. Springer, Berlin, Heidelberg, 1997.

Digilent. “Analog Discovery - USB Oscilloscope, Logic Analyzer and More.” Analog Discovery, 2020, www.analogdiscovery.com/. Accessed 6 Oct. 2020.

Learn Cryptography. “Caesar Cipher.” Accessed April 30, 2019. https://learncryptography.com/classical-encryption/caesar-cipher.

Forte, Domenic, Swarup Bhunia, and Mark M. Tehranipoor, eds. *Hardware protection through obfuscation*. Springer International Publishing, 2017.

Fournaris, A., Lidia Pocero Fraile and O. Koufopavlou. “Exploiting Hardware Vulnerabilities to Attack Embedded System Devices: a Survey of Potent Microarchitectural Attacks.” Electronics 6 (2017): 52.

Gassend, Blaise, Dwaine Clarke, Marten Van Dijk, and Srinivas Devadas. “Silicon physical random functions.” In *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 148-160. ACM, 2002.

Giraud, Christophe. “DFA on AES.” In *International Conference on Advanced Encryption Standard*, pp. 27-41. Springer, Berlin, Heidelberg, 2004.

Holcomb, Daniel E., Wayne P. Burleson, and Kevin Fu. “Power-up SRAM state as an identifying fingerprint and source of true random numbers.” *IEEE Transactions on Computers* 58, no. 9 (2009): 1198-1210.

Huang, Andrew. “Hacking the Xbox: an introduction to reverse engineering.” (2002).

Kim, Yoongu, R. Daly, Jeremie Kim, Chris Fallin, Ji-Hye Lee, Donghyuk Lee, C. Wilkerson, K. Lai and O. Mutlu. “Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors.” 2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA) (2014): 361-372.

Kocher, Paul, Joshua Jaffe, and Benjamin Jun. "Differential power analysis." In *Annual International Cryptology Conference*, pp. 388-397. Springer, Berlin, Heidelberg, 1999.

Kocher, Paul, Joshua Jaffe, and Benjamin Jun. “Differential power analysis.” In *Annual International Cryptology Conference*, pp. 388-397. Springer, Berlin, Heidelberg, 1999.

Kwong, Andrew, Daniel Genkin, D. Gruss and Yuval Yarom. “RAMBleed: Reading Bits in Memory Without Accessing Them.” 2020 IEEE Symposium on Security and Privacy (SP) (2020): 695-711.

Lipp, Moritz, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn et al. "Meltdown: Reading kernel memory from user space." In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pp. 973-990. 2018.

Maiti, Abhranil, Vikash Gunreddy, and Patrick Schaumont. "A systematic method to evaluate and compare the performance of physical unclonable functions." In *Embedded systems design with FPGAs*, pp. 245-267. Springer, New York, NY, 2013.

Majzoobi, Mehrdad, Farinaz Koushanfar, and Srinivas Devadas. "FPGA-based true random number generation using circuit metastability with adaptive feedback control." In *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 17-32. Springer, Berlin, Heidelberg, 2011.

Modchip Central. "PS2 Modbo 4.0." 2008. Accessed April 29, 2019. http://www.modchipcentral.com/store/modbo-4.0.html.

Papp, Dorottya, Zhendong Ma, and Levente Buttyan. "Embedded systems security: Threats, vulnerabilities, and attack taxonomy." In *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pp. 145-152. IEEE, 2015.

Quadir, Shahed E., Junlin Chen, Domenic Forte, Navid Asadizanjani, Sina Shahbazmohamadi, Lei Wang, John Chandy, and Mark Tehranipoor. "A survey on chip to system reverse engineering." *ACM Journal on emerging technologies in computing systems (JETC)* 13, no. 1 (2016): 6.

Robertson, Jordan, and Michael Riley. The Big Hack: The Software Side of China's Supply Chain Attack. URL: https://www.bloomberg.com/news/articles/2018-10-04/the-big-hack-the-software-side-of-china-s-supply-chain-attack. 2018

Ray, S., E. Peeters, M. Tehranipoor and S. Bhunia. “System-on-Chip Platform Security Assurance: Architecture and Validation.” Proceedings of the IEEE 106 (2018): 21-37.

Roy, Jarrod A., Farinaz Koushanfar, and Igor L. Markov. “Ending piracy of integrated circuits.” *Computer* 43, no. 10 (2010): 30-38.

Roy, Debapriya Basu, Shivam Bhasin, Sylvain Guilley, Jean-Luc Danger, Debdeep Mukhopadhyay, Xuan Thuy Ngo, and Zakaria Najm. “Reconfigurable LUT: A double edged sword for security-critical applications.” In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pp. 248-268. Springer, Cham, 2015.

Rukhin, Andrew, Juan Soto, James Nechvatal, Miles Smid, and Elaine Barker. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. Booz-Allen and Hamilton Inc Mclean VA, 2001.

Sakura Project. “SAKURA-G FPGA Board.” Satoh.Cs.Uec.Ac.Jp, 6 Jan. 2016, satoh.cs.uec.ac.jp/SAKURA/hardware/SAKURA-G.html. Accessed 6 Oct. 2020.

Schwarz, Michael, Moritz Lipp, Daniel Moghimi, Jo Van Bulck, Julian Stecklina, Thomas Prescher, and Daniel Gruss. "ZombieLoad: Cross-Privilege-Boundary Data Sampling (2019)." *arXiv preprint arXiv:1905.05726* (2019).

Suh, G. Edward, and Srinivas Devadas. “Physical unclonable functions for device authentication and secret key generation.” In *2007 44th ACM/IEEE Design Automation Conference*, pp. 9-14. IEEE, 2007.

Technologies, Terasic. “Terasic - DE Boards - Cyclone - DE1-SoC Board.” www.terasic.com.tw, Terasic Technologies, www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=836. Accessed 6 Oct. 2020.

Tehranipoor, Mohammad, and Cliff Wang, eds. Introduction to hardware security and trust. Springer Science & Business Media, 2011.

Tehranipoor, Mohammad, and Farinaz Koushanfar. “A survey of hardware Trojan taxonomy and detection.” IEEE Design & Test of Computers 27, no. 1 (2010): 10-25.

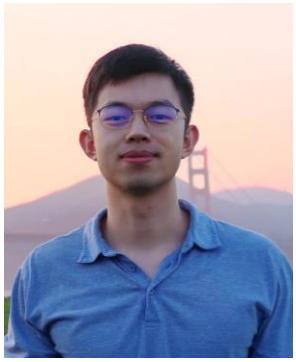
Torrance, Randy, and Dick James. “The state-of-the-art in IC reverse engineering.” In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 363-381. Springer, Berlin, Heidelberg, 2009.

Xilinx. “Spartan-7 SP701 FPGA Evaluation Kit.” Xilinx.Com, Xilinx, www.xilinx.com/products/boards-and-kits/sp701.html. Accessed 6 Oct. 2020.

Zhou, YongBin, and DengGuo Feng. “Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing.” IACR Cryptology ePrint Archive 2005 (2005): 388.

Zhang, Fengchao, Paul, Shubhra, SLPSK, Patanjali, Trivedi, Amit and Bhunia, Swarup. “On Database-Free Authentication of Microelectronic Components.” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. , no. 01 (2020): 1-13. doi: 10.1109/TVLSI.2020.3039723.

Authors



Shuo Yang graduated with a Ph.D. degree from the Department of Electrical and Computer Engineering, University of Florida, FL, USA, in 2019. He received the B.S. degree in technology and apparatus of measuring and control and the M.S. degree in instrumentation science and technology from Beihang University, Beijing, China, in 2011 and 2014. His research interests are focused on counterfeit chip detection and side-channel analysis.

Shubhra Deb Paul (Student Member, IEEE) received his B.Sc. in Electrical and Electronic Engineering from the Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2012. He earned his M.S. degree in Electrical Engineering from the Oklahoma State University, Stillwater, OK, USA, in 2016. Currently, Shubhra is pursuing a Ph.D. degree in the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. His research focuses on hardware security and trust, digital circuit simulation, and digital and embedded systems.

Swarup Bhunia (Senior Member, IEEE) received his B.E. (Hons.) from Jadavpur University, Kolkata, India, M.Tech. from the Indian Institute of Technology (IIT), Kharagpur, and Ph.D. from Purdue University, IN, USA. Currently, Dr. Bhunia is a professor and Semmoto Endowed Chair at the University of Florida, FL, USA. Earlier, he was appointed as the T. and A. Schroeder associate professor of Electrical Engineering and Computer Science at Case Western Reserve University, Cleveland, OH, USA. He has over ten years of research and development experience with over 200 publications in peer-reviewed journals and premier conferences. His research interests include hardware security and trust, adaptive nanocomputing and novel test methodologies. Dr. Bhunia received IBM Faculty Award (2013), National Science Foundation career development award (2011), Semiconductor Research Corporation Inventor Recognition Award (2009), and SRC technical excellence award as a team member (2005), and several best paper awards/nominations. He has been serving as an associate editor of IEEE Transactions on CAD, IEEE Transactions on Multi-Scale Computing Systems, ACM Journal of Emerging Technologies; served as guest editor of IEEE Design & Test of Computers (2010, 2013) and IEEE Journal on Emerging and Selected Topics in Circuits and Systems (2014). He has served in the organizing and program committee of many IEEE/ACM conferences. He is a senior member of IEEE.